IN THE CLAIMS:

1. (currently amended) For use in a wide-issue pipelined processor, a mechanism for reducing pipeline stalls between nested calls, comprising:

a program counter (PC) generator configured to generate ~~that generates~~ return PC values for multiple call instructions in a pipeline of said processor; and

return PC storage, located in an execution core of said processor, coupled to said PC generator and including a PC queue and staging registers, ~~having staging registers and located in an execution core of said processor, that stores~~ said return PC storage configured to store said return PC values in said PC queue and, upon execution of a corresponding return instruction, make one ~~instructions, makes ones~~ of said return PC values available to a PC of said processor by employing said staging registers to track said corresponding ~~ones of said~~ return instruction ~~instructions~~ while moving through stages in said pipeline.

2. (original) The mechanism as recited in Claim 1 wherein said PC generator is associated with an instruction issue unit of said processor.

3. (original) The mechanism as recited in Claim 1 wherein said PC generator generates each of said return PC values in a single clock cycle.

4. (currently amended) The mechanism as recited in Claim 1 wherein said a return PC queue ~~of said return PC storage~~ has at least as many slots as a number of call instructions that a fetch/decode stage of said pipeline can decode prior to grouping.

5. (canceled)

6. (currently amended) The mechanism as recited in Claim 1 wherein said return PC storage makes said one ~~ones~~ of said return PC values available to said a PC of said processor as said

corresponding return instruction is ~~instructions are~~ in an execution stage of said pipeline.

7. (currently amended) The mechanism as recited in Claim 1 wherein said call instructions are ~~instruction is~~ executed in a fetch/decode stage of said pipeline.

8. (original) The mechanism as recited in Claim 1 wherein said processor is a digital signal processor.

9. (currently amended) For use in a wide-issue pipelined processor, a method of reducing pipeline stalls between nested calls, comprising:

generating return PC values for call instructions in a pipeline of said processor;

storing said return PC values in a PC queue of a return PC storage having staging registers and located in an execution core of said processor; and

making one ~~ones~~ of said return PC values available to a PC of said processor upon execution of a corresponding return instruction ~~instructions~~ by employing said staging registers to track said corresponding ~~ones of said~~ return instruction ~~instructions~~ while moving through stages in said pipeline.

10. (original) The method as recited in Claim 9 wherein said generating is carried out in an instruction issue unit of said processor.

11. (original) The method as recited in Claim 9 wherein said generating comprises generating each of said return PC values in a single clock cycle.

12. (currently amended) The method as recited in Claim 9 wherein said a return PC queue of ~~said return PC storage~~ has at least as many slots as a number of call instructions that a fetch/decode stage of said pipeline can decode prior to grouping.

13. (canceled)

14. (currently amended) The method as recited in Claim 9 wherein said return PC storage makes said one ~~ones~~ of said return PC values available to said ~~a~~ PC of said processor as said corresponding return instruction is ~~instructions are~~ in an execution stage of said pipeline.

15. (currently amended) The method as recited in Claim 9 further comprising executing said call instructions ~~instruction~~ in a fetch/decode stage of said pipeline.

16. (original) The method as recited in Claim 9 wherein said processor is a digital signal processor.

17. (currently amended) A digital signal processor, comprising:

a pipeline having stages capable of executing call instructions;

a wide-issue instruction issue unit;

a program counter (PC) generator configured to generate ~~that generates~~ return PC values for multiple call instructions in a pipeline of said processor; and

return PC storage, located in an execution core of said processor, coupled to said PC generator and including a PC queue and staging registers, ~~having staging registers and located in an execution core of said processor, that stores~~ said return PC storage configured to store said return PC values in said PC queue and, upon execution of a corresponding return instruction, make one ~~instructions, makes ones~~ of said return PC values available to a PC of said processor by employing said staging registers to track said corresponding ~~ones of said~~ return instruction ~~instructions~~ while moving through stages in said pipeline.

18. (original) The DSP as recited in Claim 17 wherein said PC generator is associated with an instruction issue unit of said DSP.

19. (original) The DSP as recited in Claim 17 wherein said PC generator generates each of

said return PC values in a single clock cycle.

20. (currently amended) The DSP as recited in Claim 17 wherein said a return PC queue ~~of said return PC storage~~ has at least as many slots as a number of call instructions that a fetch/decode stage of said pipeline can decode prior to grouping.

21. (canceled)

22. (currently amended) The DSP as recited in Claim 17 wherein said return PC storage makes said one ~~ones~~ of said return PC values available to said a PC of said processor as said corresponding return instruction is ~~instructions are~~ in an execution stage of said pipeline.

23. (currently amended) The DSP as recited in Claim 17 wherein said call instructions are ~~instruction is~~ executed in a fetch/decode stage of said pipeline.

24. (new) The mechanism as recited in Claim 1 further comprising a selector configured to select and move said one of said return PC values from said PC queue to said staging registers upon said execution of said corresponding return instruction to make said one of said return PC values available for said PC.

25. (new) The mechanism as recited in Claim 24 where said selector is a multiplexer.